

# Scalable Strategies for Large-scale Structured Nonlinear Optimization

Nai-Yuan Chiang  
Mathematics and Computer Science Division  
Argonne National Laboratory

Nov, 2014

**Joint Work with: Victor Zavala**

# Motivation

- **Exploit Embedded Structures : Stochastic, PDE, Network, ...**
- **Enable Scalable and Modular Linear Algebra**

## Stochastic, Network

$$\begin{aligned} \min & \sum_{\omega \in \Omega} f_\omega(\mathbf{x}, x_\omega) \\ \text{s.t. } & g_\omega(\mathbf{x}, x_\omega) \geq 0, \omega \in \Omega \end{aligned}$$

$$K_\omega = \begin{bmatrix} H_\omega & J_\omega^T \\ J_\omega & K_\omega \end{bmatrix}$$

$$\left[ \begin{array}{ccccc} K_0 & B_1^T & B_2^T & \dots & B_\Omega^T \\ B_1 & K_1 & & & \\ B_2 & & K_2 & & \\ \vdots & & & \ddots & \\ B_\Omega & & & & K_\Omega \end{array} \right]$$

## Reduced Space (PDE)

$$\begin{aligned} \min & f(\mathbf{u}, x) \\ \text{s.t. } & g(\mathbf{u}, x) \geq 0 \\ J_x \text{ invertible} & \left[ \begin{array}{ccc} H_{xx} & H_{xu} & J_x^T \\ H_{ux} & H_{uu} & J_u^T \\ J_x & J_u & \end{array} \right] \xrightarrow{\quad} \left[ \begin{array}{cc|c} H_{xx} & J_x^T & H_{xu} \\ J_x & & J_u \\ \hline H_{ux} & J_u^T & H_{uu} \end{array} \right] \xrightarrow{\quad} \left[ \begin{array}{cc|c} J_x & J_x^T & J_u \\ H_{xx} & J_x^T & H_{xu} \\ \hline H_{ux} & J_u^T & H_{uu} \end{array} \right] \end{aligned}$$

## Stochastic PDE, Stochastic Network

$$\begin{aligned} \min & \sum_{\omega \in \Omega} f_\omega(\mathbf{x}, \mathbf{u}, x_\omega, u_\omega) \\ \text{s.t. } & g_\omega(\mathbf{x}, \mathbf{u}, x_\omega, u_\omega) \geq 0, \omega \in \Omega \end{aligned}$$

$$K_\omega = \begin{bmatrix} \Phi_0 & C_1^\top & \dots & C_n^\top \\ C_1 & \Phi_1 & & \\ \vdots & & \ddots & \\ C_n & & & \Phi_n \end{bmatrix} \text{ or } K_\omega = \begin{bmatrix} J_x & J_u^T & J_{xu} \\ H_{xx} & J_x^T & H_{xu} \\ \hline H_{ux} & J_u^T & H_{uu} \end{bmatrix}$$

$$\left[ \begin{array}{ccccc} K_0 & B_1^T & B_2^T & \dots & B_\Omega^T \\ B_1 & K_1 & & & \\ B_2 & & K_2 & & \\ \vdots & & & \ddots & \\ B_\Omega & & & & K_\Omega \end{array} \right]$$

## Stochastic PDE Network

$$\begin{aligned} \min & \sum_{\omega \in \Omega} f_\omega(\mathbf{x}, \mathbf{u}, x_\omega, u_\omega) \\ \text{s.t. } & g_\omega(\mathbf{x}, \mathbf{u}, x_\omega, u_\omega) \geq 0, \omega \in \Omega \end{aligned}$$

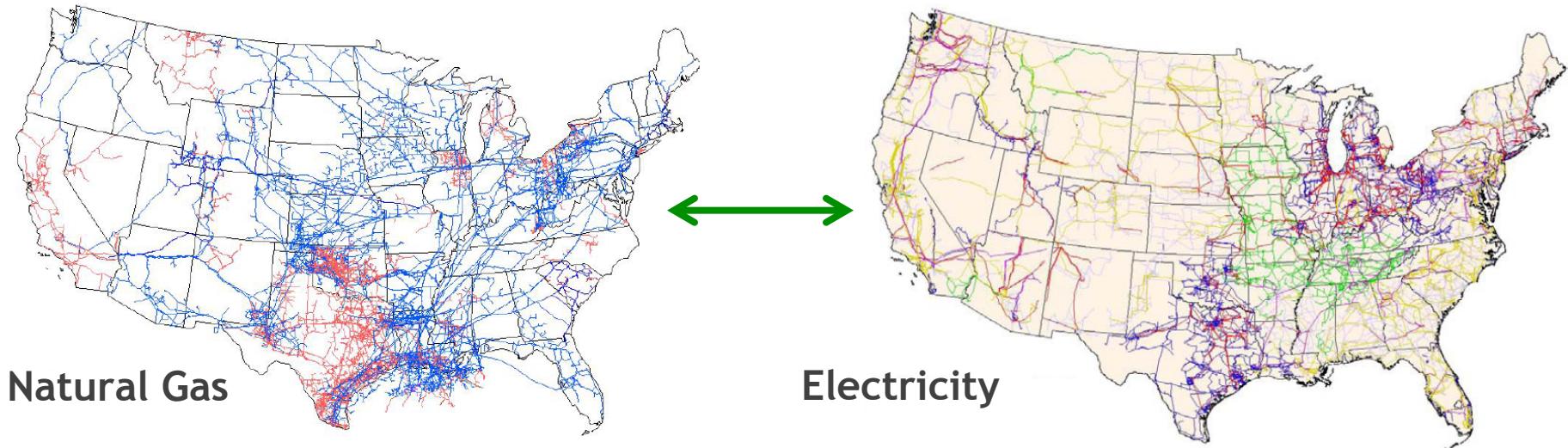
$$\Phi_\omega = \begin{bmatrix} J_x & J_u^T & J_{xu} \\ H_{xx} & J_x^T & H_{xu} \\ \hline H_{ux} & J_u^T & H_{uu} \end{bmatrix}$$

$$K_\omega = \begin{bmatrix} \Phi_0 & C_1^\top & \dots & C_n^\top \\ C_1 & \Phi_1 & & \\ \vdots & & \ddots & \\ C_n & & & \Phi_n \end{bmatrix}$$

$$\left[ \begin{array}{ccccc} K_0 & B_1^T & B_2^T & \dots & B_\Omega^T \\ B_1 & K_1 & & & \\ B_2 & & K_2 & & \\ \vdots & & & \ddots & \\ B_\Omega & & & & K_\Omega \end{array} \right]$$



# Motivation



Natural Gas

Electricity

**Practical Problems Have Embedded Structures**

## Challenges

- **How to Exploit Structures in a Scalable Manner?**
- **How to Deal With Nonconvexities in Structured Problem?**

# Outlines

Technical Motivation

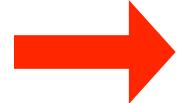
Inertia-Free Algorithm

Numerical Studies (PIPS-NLP)

# Interior-Point Framework

Nonlinear Problem (NLP):

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & c(x) = 0 \quad (\lambda) \\ & x \geq 0 \quad (\nu) \end{array}$$



Barrier Subproblem (NLP- $\mu$ )

$$\begin{array}{ll} \min & \varphi^\mu(x) := f(x) - \mu \sum_{j=1}^n \ln x_{(j)} \\ \text{s.t.} & c(x) = 0, \quad (\lambda) \end{array}$$

KKT conditions of NLP- $\mu$ :

$$\begin{aligned} \nabla_x f(x) + \nabla_x c(x)^T \lambda - \nu &= 0 \\ c(x) &= 0 \\ XVe &= \mu e \\ x, \nu &\geq 0 \end{aligned}$$

$$\begin{aligned} X &:= \text{diag}(x) \\ V &:= \text{diag}(\nu) \\ e &:= \{1, 1, \dots, 1\} \end{aligned}$$

**Augmented System:**

$$\begin{bmatrix} W_k & J_k^T \\ J_k & 0 \end{bmatrix} \begin{bmatrix} d_k \\ \lambda_k^+ \end{bmatrix} = - \begin{bmatrix} g_k \\ c_k \end{bmatrix}$$

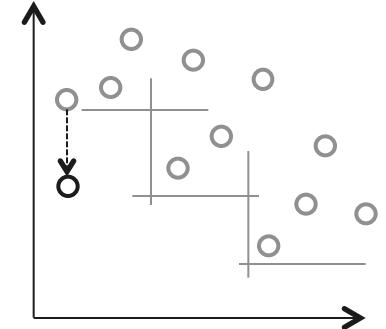
$$\begin{aligned} g_k &:= \nabla_x \varphi_k^\mu \\ J_k &:= \nabla_x c(x_k)^\top \\ H_k &:= \nabla_{xx} \mathcal{L}(x_k, \lambda_k) \\ W_k &:= H_k + \Sigma_k \\ \Sigma_k &:= X_k^{-1} V_k \end{aligned}$$

# Filter Line-Search Globalization (Waechter & Biegler, 2006)

$$\mathcal{F} := \{\theta(x), \varphi(x)\}$$

Filter Condition (FC):

$$\varphi(x) := \varphi^\mu(x)$$



Switching Condition (SC):

$$m_k(\alpha_{k,l}) < 0 \quad \text{and} \quad [m_k(\alpha_{k,l})]^{s_\varphi} [\alpha_{k,l}]^{1-s_\varphi} > \delta [\theta(x_k)]^{s_\theta}$$

where  $m_k(\alpha) := \alpha g_k^\top d_k$  is the model of the objective.

$$\theta(x) := \|c(x)\|$$

Armijo Condition (AC):

$$\varphi(x_k(\alpha_{k,l})) \leq \varphi(x_k) + \eta_\varphi m_k(\alpha_{k,l})$$

Sufficient Decrease Condition (SDC):

$$\theta(x_k(\alpha_{k,l})) \leq (1 - \gamma_\theta) \theta(x_k) \quad \text{or} \quad \varphi(x_k(\alpha_{k,l})) \leq \varphi(x_k) - \gamma_\varphi \theta(x_k)$$

**Case I:** If (SC) hold then  $d_k$  is a descent direction and we check if (AC) holds. The filter **IS NOT** updated.

**Case II:** If (SC) does not hold then we check if (SDC) holds. The filter **IS** updated.

How to ensure step is of descent under Case I?

# Step Computation Assumptions (Waechter & Biegler, 2005)

- G1: There exists an open set  $\mathcal{C}$  in which  $\varphi(x)$  and  $c(x)$  are differentiable and their function values and derivatives are bounded.
- G2:  $W_k$  are uniformly bounded for all  $k \notin R_{inc}$ .
- G3:  $W_k$  are uniformly positive definite on the null space of  $J_k$ .
- G4:  $\exists M_J > 0$  so that  $\sigma_{min}(J_k) \geq M_J$ .
- G5:  $\exists \theta_{inc} > 0$  so that  $k \notin R_{inc}$  whenever  $\theta(x_k) \leq \theta_{inc}$ .

## Lemma (Descent):

Let assumptions G hold. If  $\{x_{k_i}\}$  is a subsequence for which  $\chi_{k_i} \geq \epsilon$  with a constant  $\epsilon > 0$  independent of i, then there exists constants  $\epsilon_1, \epsilon_2 > 0$ , such that

$$\theta(x_{k_i}) \leq \epsilon_1 \implies m_{k_i}(\alpha) \leq -\epsilon_2 \alpha$$

for all i and  $\alpha \in (0, 1]$ .

# Inertia-Based Regularization (IBR)

Inertia test:

$$Inertia(M_k) = (n, m, 0)$$

$$M_k := \begin{bmatrix} W_k & J_k^T \\ J_k & \end{bmatrix}$$

If inertia test fails, regularize Hessian:

$$\begin{bmatrix} W_k + \delta_x \mathbb{I}_{n \times n} & J_k^T \\ J_k & \end{bmatrix} \begin{bmatrix} d_k \\ \lambda_k^+ \end{bmatrix} = - \begin{bmatrix} g_k \\ c_k \end{bmatrix}$$

Symmetric indefinite factorization of  $M_k := \begin{bmatrix} W_k & J_k^T \\ J_k & \end{bmatrix}$  of the form  $L B L^T$

Inertia information can be obtained from MA27, MA57, and Pardiso.

## Issues

- Different linear solvers may return **different inertia estimates**.
- Even for the same solver, different **parameter settings** may provide different estimates.
- Is difficult/impossible to get inertia from structured problems.

# Block-Bordered-Diagonal (BBD) Structure

$$\underbrace{\begin{bmatrix} K_0 & B_1^T & B_2^T & \dots & B_\Omega^T \\ B_1 & K_1 & & & \\ B_2 & & K_2 & & \\ \vdots & & & \ddots & \\ B_\Omega & & & & K_\Omega \end{bmatrix}}_M \begin{bmatrix} \Delta w_0 \\ \Delta w_1 \\ \Delta w_2 \\ \vdots \\ \Delta w_\Omega \end{bmatrix} = - \begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ \vdots \\ r_\Omega \end{bmatrix}$$
$$(K_0 - \sum B_\omega^T K_\omega^{-1} B_\omega) \Delta w_0 = -(r_0 - \sum B_\omega^T K_\omega^{-1} r_\omega) \quad \text{Schur Decomposition}$$
$$K_\omega \Delta w_\omega = -r_\omega - B_\omega \Delta w_0$$

**Inertia test:**  $\text{Inertia}(M) = (n, m, 0)$

## Haynsworth Inertia Additivity Formula

$$\text{Inertia}(M) = \text{Inertia}(K_0 - \sum B_\omega^T K_\omega^{-1} B_\omega) + \sum_{\omega \in \Omega} \text{Inertia}(K_\omega)$$

## How to Do it ( Block-Based vs. Central-Based )?

**Block-Based:** Enforce Correct Inertia for Each Block and Check Inertia of Schur Complement

**Central-Based:** Get Sum of Inertias and Perform Inertia Test

**Meaning of Inertia of A Sub-block Might Not Be Clear**

# Outlines

Technical Motivation

Inertia-Free Algorithm

Numerical Studies (PIPS-NLP)



# Inertia-Free Algorithm : Assumptions

## Step computation

Define  $d_k := n_k + t_k$  where  $J_k n_k = -c_k$  and  $t_k$  is computed from

$$\underbrace{\begin{bmatrix} W_k(\delta) & J_k^T \\ J_k & \end{bmatrix}}_{M_k(\delta)} \begin{bmatrix} t_k \\ \lambda^+ \end{bmatrix} = - \begin{bmatrix} g_k + W_k(\delta)n_k \\ 0 \end{bmatrix} \quad (*)$$

- G1: There exists an open set  $\mathcal{C}$  in which  $\varphi(x)$  and  $c(x)$  are differentiable and their function values and derivatives are bounded.
- G2:  $W_k$  are uniformly bounded for all  $k \notin R_{inc}$ .
- G3:  $W_k$  are uniformly positive definite on the null space of  $J_k$ .
- G4:  $\exists M_J > 0$  so that  $\sigma_{min}(J_k) \geq M_J$ .
- G5:  $\exists \theta_{inc} > 0$  so that  $k \notin R_{inc}$  whenever  $\theta(x_k) \leq \theta_{inc}$ .

- RG3a:

$$\exists \alpha_t > 0, t_k^T W_k(\delta) t_k + \max\{t_k^T W_k(\delta) n_k - g_k^T n_k, 0\} \geq \alpha_t t_k^T t_k$$

- RG3b:

$M_k(\delta)$  is nonsingular

# Inertia-Free Algorithm : Global Convergence

**Theorem:** Well-Defined Criticality Measure  $\Psi_k^t$  (norm of the tangential component of  $d_k$ )

Consider a subsequence  $\{x_{k_i}\}$  with  $\lim_{i \rightarrow \infty} x_{k_i} = x^*$  for a feasible  $x^*$ , let (RG3b) and (RG4) hold, let  $n_{k_i}$  satisfy  $J_{k_i} n_{k_i} = -c_{k_i}$ , and let  $t_{k_i}$  solve (\*). Then

$$\lim_{i \rightarrow \infty} \Psi_{k_i}^t = 0 \implies \lim_{i \rightarrow \infty} \|Z_{k_i}^T g_{k_i}\| = 0$$

for  $Z_{k_i}$  spanning the null space of  $J_{k_i}$ .

**Lemma:** Descent Lemma

Let (RG1)-(RG4) hold. If  $x_{k_i}$  is a subsequence of iterates for which  $\Psi_{k_i}^t \geq \epsilon$  with a constant  $\epsilon$  independent of  $i$ , then there exist positive constants  $\epsilon_1, \epsilon_2$  such that

$$\theta_{k_i} \leq \epsilon_1 \implies \frac{m_{k_i}(\alpha)}{\alpha} \leq -\epsilon_2.$$

**Theorem:** Global Convergence

Let assumptions (RG) hold. The filter line-search algorithm delivers a sequence  $\{x_k\}$  satisfying

$$\lim_{k \rightarrow \infty} \theta(x_k) = 0 \quad \text{and} \quad \liminf_{k \rightarrow \infty} \Psi^t(x_k) = 0.$$

# Alternative Inertia-Free Strategies

$$\text{RG3a: } t_k^T W_k(\delta) t_k + \max\{t_k^T W_k(\delta) n_k - g_k^T n_k, 0\} \geq \alpha_t t_k^T t_k$$

$$\begin{bmatrix} W_k + \delta_x \mathbb{I}_{n \times n} & J_k^T \\ J_k & \lambda_k^+ \end{bmatrix} \begin{bmatrix} t_k \\ \lambda_k^+ \end{bmatrix} = - \begin{bmatrix} g_k + (W_k + \delta_x \mathbb{I}_{n \times n}) n_k \\ c_k \end{bmatrix}$$

The term  $(\max\{t_k^T W_k(\delta) n_k - g_k^T n_k, 0\})$  is harmless.

→ A simpler test  $t_k^T W_k(\delta) t_k \geq \alpha_t t_k^T t_k$  is available!

It might be desirable to operate directly on the full step  $d_k$ .

$$\begin{bmatrix} W_k + \delta_x \mathbb{I}_{n \times n} & J_k^T \\ J_k & \lambda_k^+ \end{bmatrix} \begin{bmatrix} d_k \\ \lambda_k^+ \end{bmatrix} = - \begin{bmatrix} g_k \\ c_k \end{bmatrix}$$

The corresponding curvature test is:

$$d_k^T W_k(\delta) d_k + \max\{-(\lambda_k^+)^T c_k, 0\} \geq \alpha_d d_k^T d_k$$

→ Similarly, use  $d_k^T W_k(\delta) d_k \geq \alpha_d d_k^T d_k$  as the test.



# Outlines

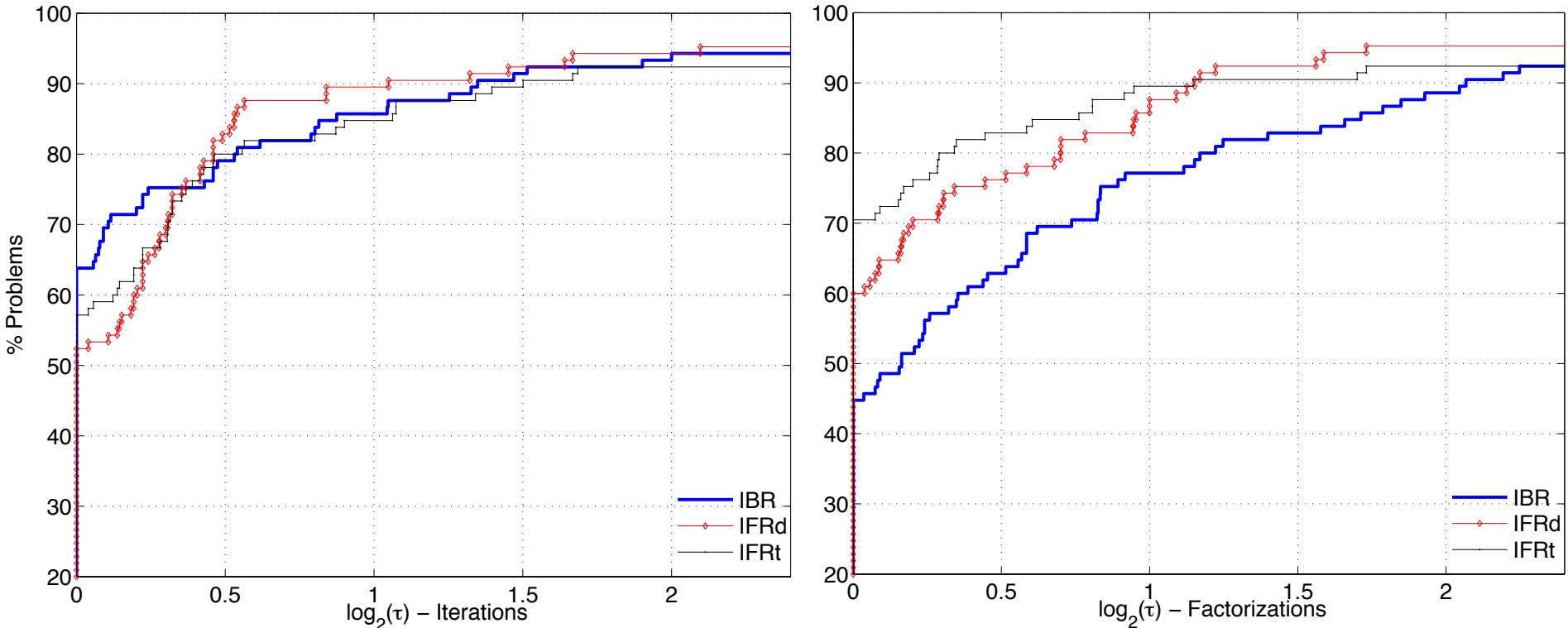
Technical Motivation

Inertia-Free Algorithm

Numerical Studies (PIPS-NLP)



# Inertia vs. Inertia-Free: CUTEr Experiments



IFRd and IFRt yield the same objective values as IBR in more than 90% of the instances.

On average, IBR requires 0.77 regularizations per iteration, while IFRd and IFRt require 0.34 and 0.24 regularizations per iteration, respectively.

# Natural Gas Optimization: Stochastic-PDE-Network

Transport Equations for link  $\ell \in \mathcal{L} := \mathcal{L}_p \cup \mathcal{L}_a$

$$\frac{\partial p_\ell}{\partial t} + \frac{1}{A_\ell} \frac{p_\ell}{\rho_\ell} \frac{\partial f_\ell}{\partial x} = 0$$

$$\frac{1}{A_\ell} \frac{\partial f_\ell}{\partial t} + \frac{\partial p_\ell}{\partial x} + \frac{8\lambda_\ell}{\pi^2 D_\ell^5} \frac{f_\ell | f_\ell|}{\rho_\ell} = 0$$

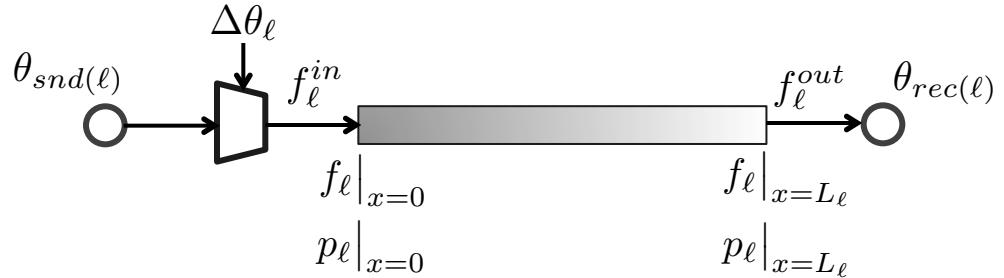
$$f_\ell|_{x=0} = f_\ell^{in}$$

$$f_\ell|_{x=L_\ell} = f_\ell^{out}$$

$$p_\ell|_{x=L_\ell} = \theta_{rec(\ell)}$$

$$p_\ell|_{x=0} = \theta_{snd(\ell)}, \ell \in \mathcal{L}_p$$

$$p_\ell|_{x=0} = \theta_{snd(\ell)} + \Delta\theta_\ell, \ell \in \mathcal{L}_a$$

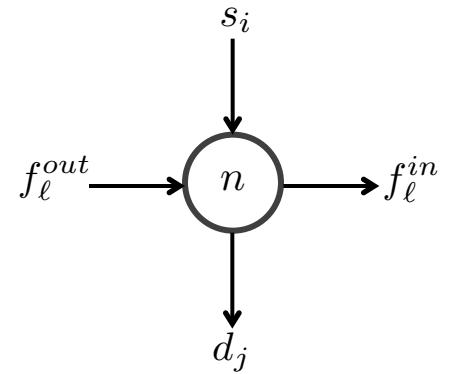


Conservation at node  $n \in \mathcal{N}$

$$\sum_{\ell: rec(\ell)=n} f_\ell^{out} + \sum_{i: sup(i)=n} s_i - \sum_{\ell: snd(\ell)=n} f_\ell^{in} - \sum_{j: dem(j)=n} d_j = 0$$

Compression Power for link  $\ell \in \mathcal{L}_A$

$$P_\ell = f_\ell^{in} c_p T \left( \left( \frac{\theta_{snd(\ell)} + \Delta\theta_\ell}{\theta_{snd(\ell)}} \right)^{\frac{\gamma-1}{\gamma}} - 1 \right)$$



# Optimal Power Flow: AC-SCOPF

(N-1) SCOPF:

Network should survive the failure of any “one” line without line-overloads.

**Kirchhoff's Voltage Law (KVL):**

$$f_{(i,j)}^P = \alpha_l v_i^2 - v_i v_j [\alpha_l \cos(\delta_i - \delta_j) + \beta_l \sin(\delta_i - \delta_j)]$$

$$f_{(i,j)}^Q = -\beta_l v_i^2 - v_i v_j [\alpha_l \sin(\delta_i - \delta_j) - \beta_l \cos(\delta_i - \delta_j)]$$

**Kirchhoff's Current Law (KCL):**

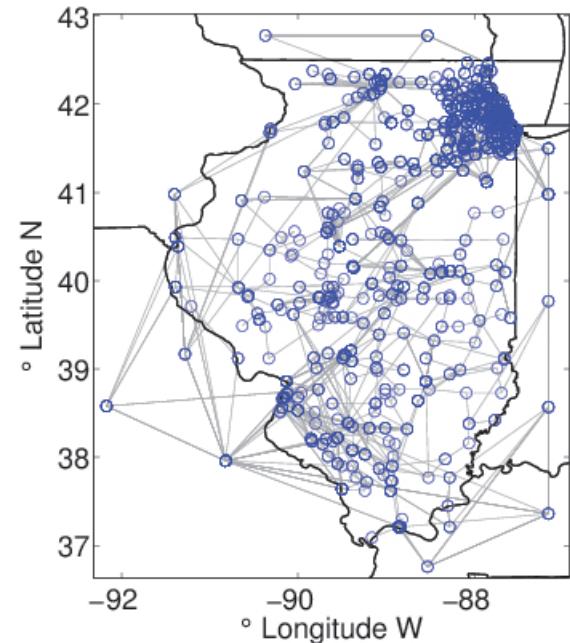
$$\sum_{g|o_g=b} p_g = \sum_{(b,i) \in L} f_{(b,i)}^P + d_b^P, \quad \forall b \in \mathcal{B}$$

$$\sum_{g|o_g=b} q_g - \beta_b v_b^2 = \sum_{(b,i) \in L} f_{(b,i)}^Q + d_b^Q, \quad \forall b \in \mathcal{B}$$

**Line Thermal Limit (limit):**

$$(f_{(i,j)}^P)^2 + (f_{(i,j)}^Q)^2 \leq (f_l^+)^2$$

$$(f_{(j,i)}^P)^2 + (f_{(j,i)}^Q)^2 \leq (f_l^+)^2$$



# Inertia vs. Inertia-Free: Larger Experiments

Problem	Domain	Algorithm	Objective	Iter	Fact
building_det	Buildings	IFR-d	$1.74 \times 10^3$	127	161
		IBR	$1.74 \times 10^3$	180	341
building_stoch_A	Buildings	IFR-d	$1.89 \times 10^3$	167	181
		IBR	$1.89 \times 10^3$	170	270
building_stoch_B	Buildings	IFR-d	$1.95 \times 10^3$	319	417
		IBR	$1.95 \times 10^3$	170	270
IEEE_162cart	Power Grid	IFR-d	$1.64 \times 10^0$	23	23
		IBR	$1.64 \times 10^0$	104	330
stochPDEgas_A	Gas Network	IFR-d	$1.73 \times 10^2$	35	35
		IBR	$1.73 \times 10^2$	34	35

Effect of Pivoting Tolerance IEEE\_162cart

ma57pivtol	Algorithm	Iterations	Factorizations
$1 \times 10^{-3}$	IFR-d	23	23
	IBR	104	226
$1 \times 10^{-4}$	IFR-d	23	23
	IBR	101	228
$1 \times 10^{-5}$	IFR-d	23	23
	IBR	179	428
$1 \times 10^{-6}$	IFR-d	23	23
	IBR	151	355



# Scalability Results for Large Network Problems

$$\begin{bmatrix} K_0 & B_1^T & B_2^T & \dots & B_\Omega^T \\ B_1 & K_1 & & & \\ B_2 & & K_2 & & \\ \vdots & & \ddots & & \\ B_\Omega & & & & K_\Omega \end{bmatrix} \quad K_\omega = \begin{bmatrix} H_\omega & J_\omega^T \\ J_\omega & \end{bmatrix}$$

$$\left( K_0 - \sum B_\omega^T K_\omega^{-1} B_\omega \right) \Delta w_0 = -(r_0 - \sum B_\omega^T K_\omega^{-1} r_\omega)$$

$$K_\omega \Delta w_\omega = -r_\omega - B_\omega \Delta w_0$$



Argonne's Fusion Cluster

## Schur Decomposition

Problem	#MPI	IBR					IFRd					IFRt				
		Obj	Iter	Fact	Time	Obj	Iter	Fact	Time	Obj	Iter	Fact	Time	Obj	Iter	Time
IEEE	16	1.36e+03	112	274	627	1.36e+03	173	190	476	1.36e+03	209	241	651			
IEEE	24	1.36e+03	112	274	424	1.36e+03	208	238	403	1.36e+03	232	255	475			
IEEE	40	1.36e+03	112	274	263	1.36e+03	160	169	181	1.36e+03	168	178	206			
IEEE	80	1.36e+03	112	274	174	1.36e+03	183	203	119	1.36e+03	177	190	127			
IEEE	120	1.36e+03	112	274	126	1.36e+03	192	224	91	1.36e+03	201	227	103			
IEEE	240	1.36e+03	113	274	65	1.36e+03	170	185	47	1.36e+03	219	240	80			
Gas	8	1.26e-02	153	278	832	1.26e-02	122	144	621	1.26e-02	93	106	491			
Gas	16	1.26e-02	136	251	363	1.26e-02	245	277	789	1.26e-02	109	122	315			
Gas	32	1.26e-02	146	274	209	1.26e-02	211	250	301	1.26e-02	99	112	143			
Gas	64	1.26e-02	157	286	123	1.26e-02	112	137	74	1.26e-02	101	114	79			
Gas	128	1.26e-02	145	275	64	1.26e-02	127	158	52	1.26e-02	109	125	52			

Performance of inertia-based and inertia-free strategies on large-scale SCOPF problems and stochastic gas network problem.

# Scalability Results for Gas Network (with No Regularization)

## Exploiting Stochastic Structure

No.Sce	n	Obj	Iters	Time(hh:mm:ss)	MPI Proc.
200	3,917,328	$1.20 \times 10^2$	55	01:01:33	20
200	3,917,328	$1.20 \times 10^2$	55	00:31:11	40
200	3,917,328	$1.20 \times 10^2$	54	00:12:36	100
200	3,917,328	$1.20 \times 10^2$	55	00:06:38	200

$$K_\omega = \begin{bmatrix} K_0 & B_1^T & B_2^T & \dots & B_\Omega^T \\ B_1 & K_1 & & & \\ B_2 & & K_2 & & \\ \vdots & & & \ddots & \\ B_\Omega & & & & K_\Omega \\ \hline H_\omega & & J_\omega^T & & \\ J_\omega & & & & \end{bmatrix}$$

## Exploiting Stochastic + PDE Structure

No.Sce	n	Obj	Iters	Time(hh:mm:ss)	MPI Proc.
200	3,917,328	$1.20 \times 10^2$	55	00:27:16	20
200	3,917,328	$1.20 \times 10^2$	55	00:14:16	40
200	3,917,328	$1.20 \times 10^2$	55	00:06:01	100
200	3,917,328	$1.20 \times 10^2$	55	00:03:14	200

$$K_\omega = \begin{bmatrix} K_0 & B_1^T & B_2^T & \dots & B_\Omega^T \\ B_1 & K_1 & & & \\ B_2 & & K_2 & & \\ \vdots & & & \ddots & \\ B_\Omega & & & & K_\Omega \\ \hline J_x & & J_u & & \\ H_{xx} & J_x^T & H_{xu} & & \\ \hline H_{ux} & J_u^T & H_{uu} & & \end{bmatrix}$$

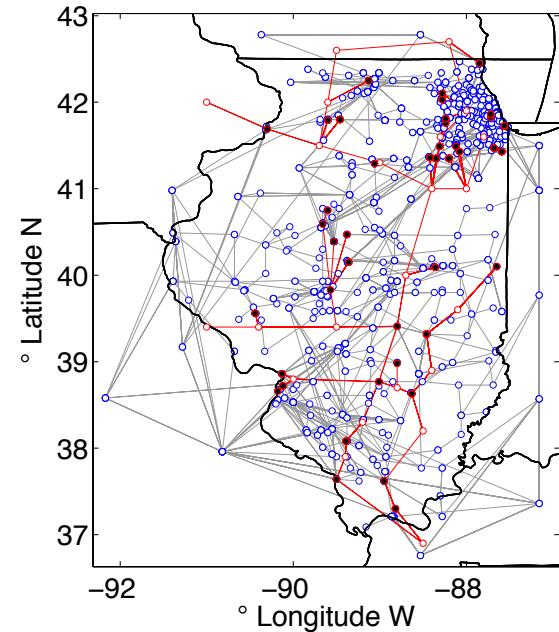
# Conclusions

## PIPS-NLP:

- Inertia-Free Algorithm (Inertia Detection for Some Classes)
- Modular Linear Algebra
- Modularity Enables:
  - Symmetric Solvers (**MA57, PARDISO**)
  - Unsymmetric and Dense Solvers (**UMFPACK, MAGMA, PETSc**)
  - Custom Solvers

## Future Work:

- Inexact Filter Line-Search
- Gas+Electricity Networks



Thank you for your attention!